1-Merges the training and the test sets to create one data set

First of all we read all data from the test and train files. According to “Daivid’s project FAQ” in discussion forum, no need to read the data in Inertial Signals file. The “subject”, “y” and “x” files are merged for each test and train, separately. Then we bind them as a data set.

2-Extracts only the measurements on the mean and standard deviation for each measurement.

We get the feature’s names from the file “features.txt”. Now we need to extract the feature name including “mean” or “std” that can be in the end or anywhere. We use function “grpl” to extract the features including “mean” and “std”. “grepl(“mean”,Names)” results a logical vector that is TRUE for a feature including “mean” and FALSE otherwise. We use which() to extract the index number of features including “mean” . We do the same for “std” and merge the extracted indices as one vector. Up to now, we have all index number of features including “mean” and “std”. Since the resulted data in task 1 has two more columns “ActivityLabel” and “subject” before feature columns, we add indices by 2. Now we easily extract the columns from the data using their index.

3- Uses descriptive activity names to name the activities in the data set

We first read the activities and their label from the file activity\_labels. We read it as a table such that has two columns called "Labels" and "Activity". This table helps us to replace the activity’s label with activity’s name. Now we merge the resulted data in task 2 with the first two column of the data in task 1. Then we use a loop for (from 1 to 6) to replace label by activity’s label.

4- Appropriately labels the data set with descriptive variable names.

We find the features including “mean” and “std” separately (using function grepl() like in task 2). Then we use function colnames to rename the columns.

5- From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

We just use plyr library and command ddply. To get the mean of columns we just use function numcolwise(mean). The resulted data has 180 rows and 81 columns.